

REMARKS

After entry of the foregoing amendment, claims 1-10 are pending in the application. Claims 8-10 are newly added.

Claims 1-7 were earlier found patentable over the art, but the March, 2006 Notice of Allowance was withdrawn to allow further prosecution.

Claim 2 has been rejected under § 112. Applicant respectfully submits that the “discourage piracy” claim language is adequately supported.

Applicant’s specification explains that the preferred embodiment redresses the failings of prior art “*fraud deterrence*” systems (see, e.g., page 3, lines 1-4).<sup>1</sup> A common definition of “*deter*” is “To prevent or *discourage* from acting.”<sup>2</sup>

“Piracy” is supported by repeated uses of terms such as “pirate,” “pirates” and “pirating” in the specification. See, e.g., page 2, line 9; page 3, line 1; page 17, line 10; and page 19, line 12.

Thus, applicant respectfully submits that – while the precise phrase “discourage piracy” is not found in the specification, the specification adequately conveys to the artisan that applicant had possession of the claimed invention.

Claims 1 and 3-7 stand rejected under § 103 over Lee (5,822,360) in view of Astrachan (5,612,974). The rejection is respectfully traversed.

Lee is concerned with methods for increasing the amount of information that can be conveyed over a channel. He notes that data compression is one technique to squeeze more data within a given bandwidth (col. 1, lines 18-20), and another is to identify areas where supplemental information can be transported with a primary signal without adversely affecting the transport of the primary signal itself (col. 1, lines 21-26). Lee’s invention concerns the latter class of techniques.

---

<sup>1</sup> See similar references at page 2, lines 16, 23 and 25.

<sup>2</sup> deter. (n.d.). *The American Heritage® Dictionary of the English Language, Fourth Edition*. Retrieved September 25, 2006, from Dictionary.com website: <http://dictionary.reference.com/browse/deter>

The June 14 Action incorrectly states:<sup>3</sup>

Lee discloses the invention (claims 1, 3, and 4) as claimed including, in a cellular telephone including a microphone, a modulator, an antenna, and an RF amplifier, the device serving to receive audio and transmit an RF signal conveying audio modulation, an improvement comprising:

Lee does not disclose a cellular telephone, nor a microphone, a modulator, an antenna, or an RF amplifier.

Indeed, the only part of Lee's disclosure that is even peripherally related to cellular telephony is his mention – at col. 8, line 12, and at col. 1, line 15, that the communications "channel" can be a wireless channel. But Lee also notes that the channel can be wired, and that his invention can be applied to audio signals recorded on magnetic or optical media or the like, such as tapes and compact discs. *See* col. 8, lines 13-38. So wireless is only a peripheral, non-essential, detail of his illustrated embodiment.

Later, the Action corrects the earlier mis-statement, conceding "Lee does not disclose a cellular telephone including a microphone, a modulator, an antenna, and an RF amplifier,"<sup>4</sup> and states that such elements are taught by Astrachan.

Astrachan is understood to teach a convolutional encoder.<sup>5</sup> As is familiar to those skilled in the art, convolutional encoding refers to an error-correcting coding technique in which an  $m$ -bit information string or symbol is transformed into an  $n$ -bit string/symbol, **where  $n > m$ .**<sup>6</sup> Thus, for example, an 80-bit message may be transformed into a 120- or 200-(or more-) bit message. (The advantage of doing so is that this expansion of the data allows it to be more reliably transferred across lossy data channels – loss of some the data bits isn't fatal to accurate data recovery.)

<sup>3</sup> June 14, 2006, Action, bottom of page 2.

<sup>4</sup> June 14, 2006, Action, top of page 3.

<sup>5</sup> C.f. title of 5,612,974: Convolutional Encoder for Use on an Integrated Circuit that Performs Multiple Communication Tasks.

<sup>6</sup> See, e.g., Wikipedia entry for "Convolution Code," submitted herewith as Exhibit A.

It will be recognized that Astrachan and Lee are at cross-purposes. Astrachan seeks *expand* the size of a given message in order to enhance reliability of reception. In doing so, he *increases* the bandwidth used. (Thus, Astrachan may take an 80-bit message and expand it so it consumes the bandwidth of a 200-bit channel.)

In contrast, Lee seeks to *squeeze* data sent across a channel in order to increase efficiency. He does this by fitting more data within a *fixed bandwidth*. (Thus, *e.g.*, Lee may take an 80-bit message and squeeze an extra 4 bits of supplemental data into it, and transmit the combined 84-bits of data over an 80-bit channel.)

Because Lee and Astrachan serve contrary purposes, an artisan reading Lee's disclosure would not be led to incorporate teachings from Astrachan, nor vice versa.

(Additionally, Lee and Astrachan are not in the same field of endeavor, so only with hindsight would any attempt to hybridize their teachings be made. Moreover, neither reference addresses the practical considerations involved in implementing a steganographic encoder within a cellular telephony device. Still further, the motivation for combination offered in the June 14 Action, *i.e.*, "*to provide a single IC that performs cellular telephone features while more readily accommodating the consumer requirements of lighter weight, smaller packages, and longer talk time,*" does not address Lee; the cited benefits are provided by Astrachan alone.<sup>7</sup>)

Accordingly, applicant respectfully submits that the claims are properly patentable over the art. Action to such end is requested.

(For brevity's sake, the foregoing discussion has noted only certain of the claims pending in the application, and only selected points have been reviewed in connection with each. Many other points that might have been raised concerning the claims, the art, and the rejections, have not been belabored.)

---

<sup>7</sup>

*Cf.*, Lee at col. 2, lines 57-61.

New claims 8-10 concern steganographically encoding digital information (e.g., audio) sent *to* a wireless portable device, recovering the steganographically encoded data at the portable device, and storing it/using it to control some aspect of the device's operation. (Support therefor is found, *e.g.*, at page 19, line 23- page 20, line 5.)

None of the art is understood to teach or suggest such arrangement.

Favorable reconsideration is solicited.

Date: October 2, 2006

**Customer Number 23735**

Phone: 503-469-4800  
FAX 503-469-4777

Respectfully submitted,

DIGIMARC CORPORATION

By \_\_\_\_\_

William Y. Conwell  
Registration No. 31,943

# Convolutional code

From Wikipedia, the free encyclopedia

In telecommunication, a **convolutional code** is a type of error-correcting code in which (a) each  $m$ -bit information symbol (each  $m$ -bit string) to be encoded is transformed into an  $n$ -bit symbol, where  $m/n$  is the code *rate* ( $n \geq m$ ) and (b) the transformation is a function of the last  $k$  information symbols, where  $k$  is the constraint length of the code.

## Contents

- 1 Where Convolutional Codes are used
- 2 Convolutional Encoding
- 3 Recursive and non-recursive codes
- 4 Impulse response, transfer function and constraint length
- 5 Trellis diagram
- 6 Free distance and error distribution
- 7 Decoding Convolutional Codes
- 8 Popular Convolutional Codes
- 9 Perforated Convolutional Codes
- 10 Design Issues
- 11 Turbo Codes: replacing Convolutional Codes
- 12 LDPC codes: replacing turbo codes
- 13 External links

### Where Convolutional Codes are used

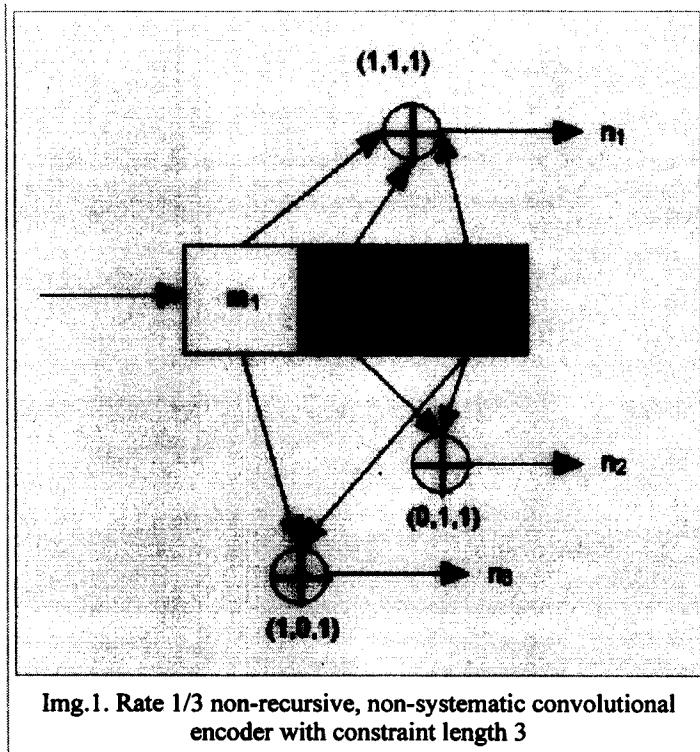
Convolutional codes are often used to improve the performance of digital radio, mobile phones and satellite links.

### Convolutional Encoding

To convolutionally encode data, start with  $k$  memory registers, each holding 1 input bit. Unless otherwise specified, all memory registers start with a value of 0. The encoder has  $n$  modulo-2 adders, and  $n$  generator polynomials -- one for each adder (see figure below). An input bit  $m_1$  is fed into the leftmost register. Using the generator polynomials and the existing values in the remaining registers, the encoder outputs  $n$  bits. Now bit shift all register values to the right ( $m_1$  moves to  $m_0$ ,  $m_0$  moves to  $m_{-1}$ ) and wait for the next input bit. If there are no remaining input bits, the encoder continues output until all registers have returned to the zero state.

The figure below is a rate 1/3 ( $m/n$ ) encoder with constraint length ( $k$ ) of 3. Generator polynomials are  $G_1 = (1,1,1)$ ,  $G_2 = (0,1,1)$ , and  $G_3 = (1,0,1)$ . Therefore, output bits are calculated as follows:

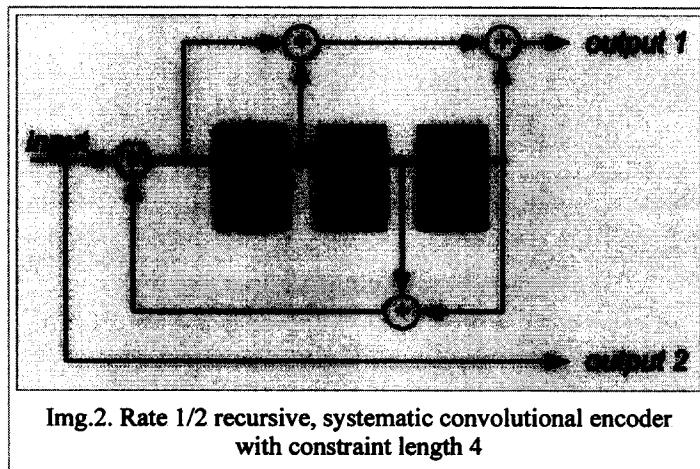
$$\begin{aligned}n_1 &= \text{mod2}(m_1 + m_0 + m_{-1}) \\n_2 &= \text{mod2}(m_0 + m_{-1}) \\n_3 &= \text{mod2}(m_1 + m_{-1})\end{aligned}$$



Img.1. Rate 1/3 non-recursive, non-systematic convolutional encoder with constraint length 3

## Recursive and non-recursive codes

The encoder on the picture above is a *non-recursive* encoder. Here's an example of a recursive one:



Img.2. Rate 1/2 recursive, systematic convolutional encoder with constraint length 4

One can see that a data sequence being encoded is here a part of the output (encoded) sequence too (look at the output 2). Such codes are referred to as *systematic* and others are called *non-systematic*.

Recursive codes are almost always systematic and, conversely, non-recursive codes are non-systematic. It isn't a strict requirement, but a common practice.

## Impulse response, transfer function and constraint length

A convolutional encoder is called so because it performs a *convolution* of the input stream with encoder's *impulse*

responses:

$$y_i^j = \sum_{k=0}^{+\infty} h_k^j x_{i-k},$$

where  $x$  is an input sequence,  $y^j$  is a sequence from output  $j$  and  $h^j$  is an impulse response for output  $j$ .

A convolutional encoder is a discrete linear time-invariant system. Every output of an encoder can be described by its own transfer function, which is closely related to a generator polynomial. An impulse response is connected with a transfer function through Z-transform.

Transfer functions for the first (non-recursive) encoder are:

- $H_1(z) = 1 + z^{-1} + z^{-2}$ ,
- $H_2(z) = z^{-1} + z^{-2}$ ,
- $H_3(z) = 1 + z^{-2}$ .

Transfer functions for the second (recursive) encoder are:

- $H_1(z) = \frac{1 + z^{-1} + z^{-3}}{1 + z^{-2} + z^{-3}}$ ,
- $H_2(z) = 1$ .

Given that maximum polynom's power (in the transfer functions) is  $m$ , *constraint length* can be defined as  $K = m + 1$ .

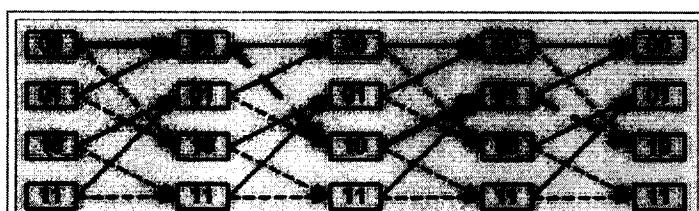
## Trellis diagram

A convolutional encoder is a finite state machine. An encoder with  $n$  binary cells will have  $2^n$  states.

Imagine that the encoder (shown on Img.1) has '1' in the left memory cell ( $m_0$ ), and '0' in the right one ( $m_{-1}$ ). ( $m_1$  is not really a memory cell because it represents a current value). We will designate such a state as "10".

According to an input bit the encoder at the next turn can convert either to "01" state or in "11" state. One can see that not all transitions are possible (e.g., a decoder can't convert from "10" state to "00" or even stay in "10" state).

All possible transitions can be shown as below:



Img.2. A trellis diagram for the encoder on Img.1

An actual encoded sequence can be represented as a path on this graph. One valid path is shown in red as an example.

This diagram gives us an idea about *decoding*: if a received sequence doesn't fit this graph, then it was received with errors, and we must choose the nearest *correct* (fitting the graph) sequence. The real decoding algorithms exploit this idea.

## Free distance and error distribution

A *free distance* ( $d$ ) is a minimal Hamming distance between different encoded sequences. A *correcting capability* ( $t$ ) of a convolutional code is a number of errors that can be corrected by the code. It can be calculated as

$$t = \left\lfloor \frac{d - 1}{2} \right\rfloor$$

Since a convolutional code doesn't use blocks, processing instead a continuous bitstream, the value of  $t$  applies to a quantity of errors located relatively near to each other. That is, multiple groups of  $t$  errors can usually be fixed when they are relatively far.

Free distance can be interpreted as a minimal length of an erroneous "burst" at the output of a convolutional decoder. The fact that errors appears as "bursts" should be accounted for when designing a concatenated code with an inner convolutional code. The popular solution for this problem is to interleave data before convolutional encoding, so that outer block (usually Reed-Solomon) code can correct most of the errors.

## Decoding Convolutional Codes

Several algorithms exist for decoding convolutional codes. For relatively small values of  $k$ , the Viterbi algorithm is universally used as it provides maximum likelihood performance and is highly parallelizable. Viterbi decoders are thus easy to implement in VLSI hardware and in software on CPUs with SIMD instruction sets.

## Popular Convolutional Codes

An especially popular Viterbi-decoded convolutional code, used at least since the Voyager program has a constraint length  $k$  of 7 and a rate  $r$  of 1/2.

- Longer constraint lengths produce more powerful codes, but the complexity of the Viterbi algorithm increases exponentially with constraint lengths, limiting these more powerful codes to deep space missions where the extra performance is easily worth the increased decoder complexity.
- Mars Pathfinder, Mars Exploration Rover and the Cassini probe to Saturn use a  $k$  of 15 and a rate of 1/6; this code performs about 2 dB better than the simpler  $k=7$  code at a cost of 256x in decoding complexity (compared to Voyager mission codes).

## Perforated Convolutional Codes

A *perforation* is a technique used to make a  $m/n$  rate code from a "basic" 1/2 code. It is reached by deletion of some bits in the decoder output. Bits are deleted according to *perforation matrix*. The following perforation matrices are the most frequently used:

code rate	perforation matrix	free distance (for NASA standard K=7 convolutional code)
1/2	1	

(No perf.)	1	10
2/3	1 0 1 1	6
3/4	1 0 1 1 1 0	5
5/6	1 0 1 0 1 1 1 0 1 0	4
7/8	1 0 0 0 1 0 1 1 1 1 1 0 1 0	3

For example, if we want to make a code with rate 2/3 using the appropriate matrix from the above table, we should take a basic encoder output and transmit every second bit from the first branch and every bit from the second one. The specific order of transmission is defined by the respective communication standard.

The perforated convolutional codes are widely used in the satellite communications, for example, in INTELSAT systems and Digital video broadcasting.

Perforated convolutional codes are also called "punctured".

## Design Issues

Longer constraint length codes are more practically decoded with any of several **sequential** decoding algorithms, of which the Fano algorithm is the best known. Unlike Viterbi decoding, sequential decoding is not maximum likelihood but its complexity increases only slightly with constraint length, allowing the use of strong, long-constraint-length codes. Such codes were used in the Pioneer program of the early 1970s to Jupiter and Saturn, but gave way to shorter, Viterbi-decoded codes, usually concatenated with large Reed-Solomon error correction codes that steepen the overall bit-error-rate curve and produce extremely low residual undetected error rates.

## Turbo Codes: replacing Convolutional Codes

Simple Viterbi-decoded convolutional codes are now giving way to turbo codes, a new class of iterated short convolutional codes that closely approach the theoretical limits imposed by Shannon's theorem with much less decoding complexity than the Viterbi algorithm on the long convolutional codes that would be required for the same performance. Turbo codes have not yet been concatenated with solid (low complexity) Reed-Solomon error correction codes. However, in the interest of planetary exploration this may someday be done.

## LDPC codes: replacing turbo codes

The low-density parity-check code is being adopted to replace turbo coding, giving slightly better performance at the cost of more computational complexity.

## External links

- Tutorial on Convolutional Coding and Decoding (<http://complextoreal.com/chapters/convo.pdf>)

Retrieved from "[http://en.wikipedia.org/wiki/Convolutional\\_code](http://en.wikipedia.org/wiki/Convolutional_code)"

---

- This page was last modified 10:49, 23 September 2006.
- All text is available under the terms of the GNU Free Documentation License. (See **Copyrights** for details.)

Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc.